

Oracle® Hyperion Enterprise Performance Management Architect

Batch Client User's Guide

リリース 11.1.2.3

Performance Management Architect Batch Client User's Guide, 11.1.2.3

Copyright © 2006, 2013, Oracle and/or its affiliates. All rights reserved.

著者: EPM 情報開発チーム

Oracle および Java は Oracle Corporation およびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT RIGHTS:

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

このソフトウェアもしくはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアもしくはハードウェアは、危険が伴うアプリケーション（人的傷害を発生させる可能性があるアプリケーションを含む）への用途を目的として開発されていません。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用する際、安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用したことにより起因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

目次

ドキュメントのアクセシビリティについて	5
第 1 章 Performance Management Architect バッチ・クライアントの使用方法	7
バッチ・クライアントの起動	8
コマンド・ライン・オプション	8
戻りコード	10
ロギング	12
第 2 章 コマンド・ファイルの構成	15
変数	16
コメント	17
コマンド	18
スクリプト	19
一般的に使用されるコマンド	19
Option コマンド	20
Login / Logout コマンド	20
login	20
Logout	21
Quit	22
Exit	22
Copy コマンド	22
Application	22
Dimension	22
Create コマンド	23
Application	23
Dimension	24
Member	24
Create Association コマンド	25
Delete Association コマンド	25
Delete コマンド	26
Application	26
Association	26
Dimension	26

Member	27
Detach Dimension コマンド	27
Exclude コマンド	28
Member	28
Execute コマンド	29
インポート(アプリケーションまたは共有ライブラリへ)	29
次元の同期(共有ライブラリとの)	30
配置	30
再配置	32
データの同期	33
検証	34
Include コマンド	35
Dimension	35
Insert コマンド	36
Insert Member	36
移動コマンド	36
メンバーの移動	36
Rename コマンド	37
メンバー名の変更	37
Remove コマンド	37
Remove Member (アプリケーションまたは共有ライブラリ)	37
Remove Dimension	38
Share Dimension コマンド	38
Update コマンド	39
Application	39
Dimension	40
Member	40
次元の関連付け	40
索引	43

ドキュメントのアクセシビリティについて

Oracle のアクセシビリティについての詳細情報は、Oracle Accessibility Program の Web サイト <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc> を参照してください。

Access to Oracle Support

Oracle サポート・サービスでは、My Oracle Support を通して電子支援サービスを提供しています。詳細情報は <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> か、聴覚に障害のあるお客様は <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> を参照してください。

1

Performance Management Architect バッチ・クライアントの使用法

この章の内容

バッチ・クライアントの起動	8
コマンド・ライン・オプション	8
戻りコード	10
ロギング	12

Oracle Hyperion EPM Architect バッチ・クライアントによって、データのエクスポート、メタデータのロード、データのロード、計算などのプロセスを組み合わせ、通常の夜間または週末のロード・プロセスの処理中に、これらの操作を開始できます。

バッチ・クライアントによる外部スケジューリング・ツールを使用してプロセスを開始できます。バッチ・クライアントを使用すると、次のような多くのタスクを実行できます:

- Performance Management Architect にメタデータをロードする
- 次元およびメジャーのセキュリティに関するプロパティを更新する
- アプリケーションにデータをロードする

バッチ・クライアントは、Performance Management Architect をインストールするときに自動的にインストールされます。バッチ・クライアントは、Windows プラットフォームで実行できます。Performance Management Architect をインストールすると、インストール中に生成されるクラス・パスを設定するためのバッチ・ファイルが自動的に作成されます。

Performance Management Architect バッチ・クライアントは、次の 2 つのモードで実行できます:

- コマンド・ライン・モード
 - コマンドを対話的に入力できる
 - 各コマンドは、複数の行にまたがることできる
 - 複数のコマンド・ステートメントをセミコロン(;)で区切ることができる
 - 単一引用符を使用する必要がある
 - コマンドは即座に実行される
- スクリプト・モード
 - 対話なしで一連のコマンドを実行する

- プログラムの起動時にコマンド・ファイルを指定でき、オプションで結果ログ・ファイルを指定できる
- サードパーティのスケジューラを使用して、スクリプトの実行をスケジュールできる

バッチ・クライアントの起動

Performance Management Architect バッチ・クライアントは、対話型のコマンド・ライン・モードで実行するか、またはコマンド・ラインで指定するスクリプト・ファイルを実行できます。

- ▶ バッチ・クライアントを起動するには、「スタート」、「プログラム」、「Oracle EPM System」、「Foundation Services」、「Performance Management Architect」、「EPMA バッチ・クライアントの起動」の順に選択します。

EPM_ORACLE_HOME\products\Foundation\BPMA\EPMABatchClient に移動し、対話型コマンド・ラインまたはスクリプトを使用して epma-batch-client.bat ファイルを起動することもできます。(LINUX または UNIX 環境で、EPM_ORACLE_HOME/products/Foundation/BPMA/EPMABatchClient に移動して epma-batch-client.sh ファイルを起動します。)

たとえば、対話型コマンド・ライン・モードでバッチ・クライアントを起動する場合は、次のようにパラメータなしでバッチ・ファイルを起動します:

```
EPM_ORACLE_HOME\products\Foundation\BPMA\EPMABatchClient\epma-batch-client.bat
```

スクリプトを起動する場合は、スクリプト・ファイル名に -C オプションを付けて指定する必要があります。すべての追加パラメータは任意です。例:

```
EPM_ORACLE_HOME\products\Foundation\BPMA\EPMABatchClient\epma-batch-client.bat -CMyScript.txt
```

注: スクリプト・ファイル名とログ・ファイル名は、相対パスで指定できますが、バッチ・ファイルを起動するフォルダではなく、EPMABatchClient フォルダに対する相対パスにする必要があります。

コマンド・ライン・オプション

バッチ・クライアントを起動するときは、次の引数を指定できます。

表 1 バッチ・クライアントのコマンド

コマンド	説明
-H	バッチ・クライアントのヘルプを表示します。
-C	実行するスクリプト・ファイルの名前を指定します。 例: -C'C:\Scripts\LightsOut.txt'

コマンド	説明
-G	<p>使用する言語を指定します。このパラメータを使用するには、-Gx と入力します。ここで、x は、次の言語コードのいずれかです:</p> <ul style="list-style-type: none"> ● DA - デンマーク語 ● DE - ドイツ語 ● ES - スペイン語 ● FR - フランス語 ● IT - イタリア語 ● JA - 日本語 ● KO - 韓国語 ● PT_BR - ブラジル・ポルトガル語 ● RU - ロシア語 ● SV - スウェーデン語 ● TR - トルコ語 ● ZH_CN - 簡体字中国語 ● ZH_TW - 繁体字中国語 <p>たとえば、フランス語の場合は-Gfr、イタリア語の場合は-Git となります。</p> <p>-G オプションが指定されていない場合、バッチ・クライアントはオペレーティング・システムの現在のデフォルト言語を使用しようとします。リソース・ファイルが見つからない場合は、デフォルト言語である英語が使用されます。特定の言語のリソース・ファイルで文字列が見つからない場合は、英語版が使用されます。</p>
-R	<p>マシンに結果を書き込むファイルの名前を指定します。</p> <p>例: -R'C:\LogFiles\ScriptResult.log'</p>
-L	<p>コマンドをログ出力するかどうかを指定します。デフォルト値は 0 です。</p> <p>ログのコマンドは次のとおりです:</p> <p>Off (デフォルト) = コマンドをログ出力しない</p> <p>On = コマンドをログ出力する</p> <p>例: -LOn</p>
-S	<p>コマンドが失敗すると、スクリプトの実行を停止します。デフォルト値は 1(TRUE) です。</p> <p>エラーが発生した場合は停止: TRUE</p> <p>エラー時に続行: FALSE</p> <p>例: -SFalse エラーが発生しても続行します。</p>
-U	<p>Performance Management Architect にログインするために使用するユーザー名を指定します。</p> <p>例: -U' Admin'</p>
-P	<p>Performance Management Architect にログインするために使用するパスワードを指定します。</p> <p>例: -Ppassword</p>

コマンド	説明
-V	スクリプト確認のオン/オフを切り換えます。検証をオンにすると、実行前にスクリプトの構文エラーがチェックされます。エラーがあると、スクリプトの実行は停止します。「Off」を指定すると、実行前に検証は行われません。「On」を指定すると、実行前にスクリプトが検証されます。デフォルト値「On」で、実行前にスクリプトが検証されます。たとえば、 <code>-Voff</code> を指定すると、実行前にスクリプトは検証されません。
-0	検証のみです。これを指定すると、バッチ・クライアントはスクリプトを検証しますが、結果にかかわらず、スクリプトは実行されません。これは、スクリプトに構文エラーがあるかどうかをテストするために使用できます。
-N	コンソールに表示される出力に使用する文字エンコードを指定します例: <code>-Ncp866</code> Windows 環境では、デフォルトはロケールのコード・ページ(<code>chcp</code> コマンドの使用を決定できる)です。 UNIX 環境では、デフォルトは UTF-8 です。 注： 日本語の Windows 環境では、コード・ページを MS932 に設定する必要があります。例: <code>-NMS932</code> 。

注： `-S` などの On/Off を使用するコマンド・ライン・パラメータでは、0 または 1、Y または N、True または False、On または Off を使用できます。たとえば、`-S0`、`-SY`、`-SNo`、`-STrue`、`-SOff` などは、すべて有効です。

次に、コマンド・ライン・オプションの使い方のコード例を示します。

```
epma-batch-client -H
epma-batch-client -C"C:\EPM_ORACLE_INSTANCE\EPMA\Commands.txt"
-R"C:\EPM_ORACLE_INSTANCE\EPMA\ResultFile.txt" -LOn -SFalse
-Uadmin -Ppassword
```

戻りコード

バッチ・クライアントが終了するときは、スクリプトの終了状態に基づいて呼び出しプログラムに結果コードが戻されます。StopOnError の設定に基づいて、次の 2 つのシナリオがあります。

StopOnError = False (-S0)

StopOnError が FALSE の場合の戻りコードは、一般的な成否を表します。

0 = 成功(エラーなし)

-1 = 失敗(1 つ以上のエラーが発生)

StopOnError = TRUE (-S1)

StopOnError が TRUE の場合、エラーが発生するとバッチ・クライアントは終了し、次の表に基づくコードを戻します。デフォルトでは、StopOnError は TRUE に設定されています。

バッチ・クライアントは、成否に基づいて次のような結果コードを戻します。

コマンド	コマンド・コード	クラス	クラス・コード	戻りコード
成功	N/A	N/A	N/A	0
一般エラー	N/A	N/A	N/A	-1
検証エラー	N/A	N/A	N/A	1
解析エラー	N/A	N/A	N/A	100
コマンド・ライン・エラー	N/A	N/A	N/A	4
Copy	15	Application	1	1501
Copy	15	Dimension	2	1502
Create	1	Application	1	101
Create	1	Dimension	2	102
Create	1	Member	3	103
Create	1	Association	10	110
Debug	21	N/A	N/A	2100
Delete	2	Application	1	201
Delete	2	Dimension	2	202
Delete	2	Member	3	203
Delete	2	Association	10	210
Detach	16	Dimension	2	1602
Exclude	3	Member	3	303
Execute	4	DataSynchronization	4	404
Execute	4	Deploy	5	405
Execute	4	DimensionSynchronization	6	406
Execute	4	Import	7	407
Execute	4	Redeploy	9	409
Execute	4	Validate	12	412
Exit	5	N/A	0	500
Include	6	Dimension	2	602
Include	6	Member	3	603
Insert	7	Member	3	703

コマンド	コマンド・コード	クラス	クラス・コード	戻りコード
Login	8	N/A	0	800
Logout	9	N/A	0	900
Move	19	Member	3	1903
Option	20	N/A	0	2000
Quit	10	N/A	0	1000
Remove	11	Dimension	2	1102
Remove	11	Member	3	1103
Rename	18	Member	3	1803
Set	12	N/A	0	1200
Share	17	Dimension	2	1702
Update	13	Application	1	1301
Update	13	DimensionAssociation	8	1308
Update	13	Dimension	2	1302
Update	13	Member	3	1303
Variable	14	N/A	0	1400

DOS のバッチ・ファイルまたは Windows のコマンド・ファイルで、次のようにして、エラーをチェックできます:

```
Call epma-batch-client.bat .\scripts\MyScript.txt
IF ERRORLEVEL 0 goto ON_SUCCESS
IF ERRORLEVEL 100 goto PARSE_ERROR
If ERRORLEVEL 101 goto APP_CREATE_FAILED
```

ロギング

バッチ・クライアントには、結果ファイルとトレース・ファイルを使用して出力できるいくつかのロギングのレベルがあります。結果ファイルには、コマンド・ウィンドウに入力するコマンドの詳細およびその実行ステータス、エラーまたは警告メッセージが含まれます。トレース・ファイルには、デバッグに役立つスタック・トレースの詳細が含まれます。

バッチ・クライアントでは、ロギングに Oracle Diagnostics Logging (ODL) を使用します。すべての構成パラメータが conf ディレクトリの logger.xml に格納されます。たとえば、EPM_ORACLE_INSTANCE\products\Foundation\BPMA\EPMABatchClient\output などです。

次に、logger.xml ファイルのコード例を示します。

```

<logging_configuration>
<log_handlers>
  <log_handler name="traceFile" class="oracle.core.ojdl.logging.ODLHandlerFactory">
    <property name="path" value="{user.dir}/output/EPMABatchClientTrace.log"/>
    <property name="maxFileSize" value="1000000"/>
    <property name="maxLogSize" value="5000000"/>
    <property name="useSourceClassAndMethod" value="true"/>
    <property name="encoding" value="UTF-8"/>
  </log_handler>
  <log_handler name="console"
    level="ALL"
    class="java.util.logging.ConsoleHandler"
    formatter="com.hyperion.bpma.logger.GenericFormatter"/>
  <log_handler name="resultsFile"
    class="com.hyperion.bpma.logger.GenericFileHandler"
    formatter="com.hyperion.bpma.logger.GenericFormatter"
    level="ALL"/>
</log_handlers>
<loggers>
  <logger name="BatchClient.trace" level="ALL" useParentHandlers="false">
    <handler name="traceFile"/>
  </logger>
  <logger name="BatchClient.console" level="INFO" useParentHandlers="false">
    <handler name="console"/>
  </logger>
  <logger name="BatchClient.results" level="INFO" useParentHandlers="false">
    <handler name="resultsFile"/>
  </logger>
</loggers>
</logging_configuration>

```

コマンドが正常終了したかどうかを簡単に判断できるように、次のコマンドについて、関連するジョブ ID およびジョブ ID URL がログに記録されます。

- Execute Deploy
- Execute ReDeploy
- Execute Validate
- Execute DataSynchronization
- Execute Import
- Copy Application
- Detach Dimension
- Share Dimension

注： Java 1.6 で使用可能な標準フォーマッタを使用するオプションがあります。たとえば、`com.hyperion.bpma.logger.GenericFormatter` のかわりに `java.util.logging.SimpleFormatter` を使用できます。

2

コマンド・ファイルの構成

この章の内容

変数	16
コメント	17
コマンド	18
スクリプト	19
一般的に使用されるコマンド	19

コマンド・ファイルは、バッチ・クライアントの入力ファイルです。これには、次のうちの1つ以上が含まれている可能性があります:

- コマンド
- 変数、宣言および割当て
- コメント

次に、コマンド・ファイルのコード例を示します。

```
// Test Script
set bpmaserverurl=http://localhost/hyperion-bpma-server;
set workspaceurl=http://localhost:19000/workspace;

login admin,password;

set ApplicationName = 'Sample';

// Delete some members
Delete Member
    Properties(MemberName, DimensionName, ParentName, DeleteAllDescendants)
    Values('M1-1-1', 'A1', 'M1-1', true);

Delete Member
    Properties(MemberName, DimensionName, ParentName, DeleteAllDescendants)
    Values('M1', 'A1', '#root', false);

Delete Member
    Properties(MemberName, DimensionName, ParentName, DeleteAllDescendants)
    Values('M1', 'A1', '#root', true);

Delete Dimension
    Properties(DimensionName)
    Values('A1');
```

```

Delete Dimension
  Properties(DimensionName)
  Values('E1');

Delete Application
  Properties(ApplicationName, WaitForCompletion)
  Values('TestApp1');

set ApplicationName = '';

// Delete shared dims
Delete Dimension
  Properties(DimensionName)
  Values('S1');

quit;

```

コマンド・ファイルのほとんどのコマンドは即座に実行されます。ただし、EXECUTE コマンドは実行に時間がかかり、WaitForCompletion パラメータをサポートします。DIMSYNCRONIZATION を除くすべて実行コマンドはこのパラメータをサポートします。たとえば、WaitForCompletion パラメータを使用して、バッチ・クライアントにコマンドが実行されるまで待機させることもできます。次のコマンドに WaitForCompletion パラメータの例を示します。この場合、管理者は実行時間が比較的長いデータ同期コマンドを実行しています。実行に長い時間を要するその他のコマンドとしては、インポートおよびアプリケーションの配置があります。

```

execute datasynchronization
  parameters(DataSynchronizationName, DataTransformationOperator,
DataTransformationValue, FileName, ValidateOnly, WaitForCompletion)
  values('CommaSync3', '*', '1.2345', '', 'false', 'true');

```

変数

変数は、定義してスクリプトで呼び出すことができます。変数の特性は次のとおりです:

- 変数には、名前と単一の値のタイプがある
- 変数の値は、任意のデータ型にできる
- 一度定義した変数を複数の場所で使用できる
- コマンド間で変数の値を変更できる
- 変数名には大文字と小文字の区別がある
- var キーワードを使用して変数を定義し、\$を使用して変数を参照できる

次に、変数を使用して共有次元を作成するコード例を示します。

```
// Create Shared Dimension Script
```



```
set bpmaserverurl=http://localhost/hyperion-bpma-server;
set workspaceurl=http://localhost:19000/workspace;

login admin,password;
var DimType='Scenario';
// Create a shared dimension
create Dimension
  Properties(DimensionName, DimensionDescription, DimensionType)
  Values('S1', 'New Scenario', '$DimType');
```

次の3つのタイプの変数があります:

- ユーザー

ユーザー変数は、var キーワードを使用して割り当てます。例:

```
var variable1 = 'abc';
```

- システム

システム変数は、set コマンドを使用して割り当てます。例:

```
set bpmaserverurl='http://localhost/hyperion-bpma-server';
set workspaceurl='http://localhost:19000/workspace';
```

- オブジェクト

オブジェクト変数は、set コマンドを使用して割り当てます。例:

```
set ApplicationName = 'Comma';

set dimension=Account;
```

コメント

行の先頭に2つのスラッシュ(//)を記述すると、スクリプト内の任意の行をコメントにできます。複数の行にわたってコメントにする場合は、各行をコメントにする必要があります。例:

```
//execute datasynchronization
// parameters(DataSynchronizationName, DataTransformationOperator,
DataTransformationValue, FileName,
//   ValidateOnly, WaitForCompletion)
// values('CommaSync3', '*', '1.2345', '', 'false', 'true');
```

コマンド

コマンドの構成要素は次のとおりです:

- コマンド動詞
- コマンド・クラス
- パラメータ値の集合またはプロパティ値の集合

パラメータおよびその値は、カンマによって区切ります。サポートされているコマンド動詞は、次のとおりです:

- CREATE
- COPY
- DEBUG
- DELETE
- DETACH
- EXCLUDE
- EXECUTE
- EXIT
- INCLUDE
- INSERT
- LOGIN
- LOGOUT
- MOVE
- OPTION
- QUIT
- REMOVE
- RENAME
- SHARE
- UPDATE
- SET
- VARIABLE

サポートされているコマンド・クラスは、次のとおりです:

- APPLICATION
- ASSOCIATION
- DIMENSION
- DIMENSIONASSOCIATION
- DIMSYNCHRONIZATION

- MEMBER
- IMPORT
- DEPLOY
- DATASYNCHRONIZATION
- REDEPLOY
- VALIDATE

ヒント: 次元の root メンバーには、#root 定数を使用します。共有ライブラリを宛先にするコマンドのアプリケーション名には、#shared 定数を使用します。

スクリプト

スクリプトは、連続して実行できるコマンドの集合です。スクリプトには、任意の順序でコマンドを記述できますが、login のような特定の初期化コマンドは、他のコマンドより前に実行する必要があります。スクリプト内の各コマンドはセミコロン(;)によって区切られます。コマンドには空白文字を含め、複数の行にまたがれます。

一般的に使用されるコマンド

サブトピック

- [Option コマンド](#)
- [Login / Logout コマンド](#)
- [Copy コマンド](#)
- [Create コマンド](#)
- [Create Association コマンド](#)
- [Delete Association コマンド](#)
- [Delete コマンド](#)
- [Detach Dimension コマンド](#)
- [Exclude コマンド](#)
- [Execute コマンド](#)
- [Include コマンド](#)
- [Insert コマンド](#)
- [移動コマンド](#)
- [Rename コマンド](#)
- [Remove コマンド](#)
- [Share Dimension コマンド](#)
- [Update コマンド](#)

次の項では、一般的に使用されるコマンドの例を示します。

Option コマンド

Option コマンドを使用すると、スクリプトの実行中にコマンド・ライン・オプションを動的に変更できます。変更可能なオプションは次のとおりです:

```
StopOnError
option StopOnError = true;
```

```
EchoComments
option EchoComments = true;
```

```
LogCommands
option LogCommands = true;
```

Login / Logout コマンド

サブトピック

- [login](#)
- [Logout](#)
- [Quit](#)
- [Exit](#)

login

Performance Management Architect にログインします。

スクリプトを使用してログインする場合は、次のように入力します:

```
Login admin,password;
```

コマンド・ラインを使用してログインする場合は、次のように入力します:

```
Login;
```

プロキシ経由でのログイン

SET コマンドを使用し、次の変数を定義して、プロキシ経由でログインできます。

```
set ProxyHost='http://localhost/myProxy';
set ProxyPort='8080';
set ProxyUsername='myProxyUser';
```

```
set ProxyPassword='myProxyPassword';
```

ProxyHost - プロキシ・ホスト名。

ProxyPort - プロキシ・ポート番号。

ProxyUsername - 認証するユーザー名。

ProxyPassword - ユーザー・パスワード。

Oracle Access Manager などのシングル・サインオン(SSO)を使用したログイン

シングル・サインオンを使用したログインには2つのオプションがあります:

- 基本/ダイジェスト認証

基本/ダイジェスト認証の場合は、次のように SSOType を Basic に設定します:

```
set SSOType='Basic';
```

事前認証(最適化に使用)の場合は、AuthenticationPreemptive を次のように設定します:

```
set AuthenticationPreemptive='True';
```

- フォーム・ベース認証

フォーム・ベース認証の場合は、次のように SSOType を Form に設定します:

```
set SSOType='Form';
```

フォーム・ベース認証を使用している場合は、ユーザー名とパスワードをポストするフィールド名を次のように定義する必要があります:

```
set SSOFormUsernameField='userid';  
set SSOFormPasswordField='password';
```

注: デフォルトのフィールド名は'userid'および'password'です。

Logout

Performance Management Architect からログアウトします。

```
Logout;
```

Quit

バッチ・クライアントを閉じます。

```
Quit;
```

Exit

バッチ・クライアントを閉じます。

```
Exit;
```

Copy コマンド

サブトピック

- [Application](#)
- [Dimension](#)

Application

Copy Application コマンドは、アプリケーション・ライブラリで使用可能な"Duplicate As New"コマンドと同じです。

```
Copy Application
Properties ( ApplicationName, CopyApplicationToName, ApplicationDescription,
ApplicationType)
Values( 'Comma', 'CommaCopy', 'Copied App Desc', 'Consolidation');
```

ApplicationName - 既存のアプリケーションの名前です。

CopyApplicationToName - 新規複製されたアプリケーションの名前です。

ApplicationDescription - アプリケーションの説明です。

ApplicationType - アプリケーションのタイプです。有効な値は次のとおりです: 汎用、連結、Planning、Profitability-標準、Profitability-詳細、Essbase ASO または Essbase BSO。

Dimension

共有ライブラリ内、アプリケーション内または共有ライブラリとアプリケーションの間で次元をコピーする方法を提供します。あるアプリケーションから別のアプリケーションに次元を直接コピーできません。

```
Copy Dimension
Properties(ApplicationName, DimensionName, TargetDimensionName,
```

```
TargetDimensionDescription,  
destApplicationName)  
  Values('#Shared', 'Scenario','CopyScenario', 'Copy of Scenario Dim', Comma');
```

ApplicationName - 既存のアプリケーションの名前です。

DimensionName - 既存の次元の名前です。

TargetDimensionName - ターゲット次元の名前です。

TargetDimensionDescription - ターゲット次元の説明です。

destApplicationName - 宛先アプリケーションの名前です。

Create コマンド

サブトピック

- [Application](#)
- [Dimension](#)
- [Member](#)

Application

指定された名前で空のアプリケーションを新規作成します。

```
Create Application  
Properties(ApplicationName, ApplicationDescription, ApplicationType)  
Values('Comma', 'Description for Comma', 'Consolidation');
```

ApplicationName - アプリケーションの有効な名前を含む文字列です。

ApplicationDescription - アプリケーションの有効な名前を含む文字列です。

ApplicationType - 次の値をサポートします:

- 汎用
- 連結
- Planning
- Enterprise Analytics
Essbase ASO アプリケーションの場合
- Essbase Analytics
Essbase BSO アプリケーションの場合
- Profitability
- Profitability-詳細

Dimension

指定されたアプリケーションまたは共有ライブラリに空の次元を新規作成します。

```
Create Dimension
Properties(ApplicationName, DimensionName, DimensionDescription,
DimensionType)
Values('Comma', 'Test_Account', 'Test Account', 'Account');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリに次元を作成する場合は、`#Shared` を使用します。

DimensionName - 次元の有効な名前です。

DimensionType - 次元タイプは、次のいずれかを使用できます:

- 勘定科目
- 別名
- AllocationType
- 属性
- ConsolidationMethod
- 国
- 通貨
- エンティティ
- 汎用
- ICP
- メジャー
- MeasuresDetailed
- シナリオ
- SecurityClass
- スマート・リスト
- 時間
- UDA
- 値
- バージョン
- 表示
- 年

Member

指定された次元に新規メンバーを作成します。

```
Create Member
```



```
Properties(ApplicationName, DimensionName, ParentName, MemberName, MemberDescription)
Values('Comma', 'Member_Dim', '#root', 'TestMember1', 'Description for TestMember1');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリに次元を作成する場合は、#Shared を使用します。

DimensionName - 既存の次元の名前です。

ParentName - 新たに作成したメンバーを挿入する親の名前です。ツリーのトップ・レベルに新規メンバーを追加する場合は、#Root を使用します。

MemberName - 新規メンバーの有効な名前です。

MemberDescription - 新規メンバーの説明です。

Create Association コマンド

2 つの次元の間に関連付けを作成します。基本次元を共有する場合は、関連次元を共有次元にする必要があります。

```
Create Association
Properties(ApplicationName, DimensionName, AssociatedDimensionName,
PropertyDescription)
Values('Comma', 'Scenario', 'AttribDim', 'AttribProp', 'Attrib Prop Desc');
```

ApplicationName - 既存のアプリケーションの名前です。

DimensionName - 既存の次元の名前です。

AssociatedDimensionName - 関連付ける次元の名前です。

PropertyDescription - 更新するプロパティの名前です。必要なプロパティをいくつでも記載できますが、プロパティには有効な値を指定する必要があります。

PropertyDescription - コメントまたはプロパティの説明を入力できるオプションのパラメータ。

Delete Association コマンド

既存の次元の関連付けを削除します。

```
Delete Association
Properties(ApplicationName, DimensionName, PropertyName)
Values('Comma', 'Scenario', 'SecurityClass');
```

ApplicationName - 既存のアプリケーションの名前です。

DimensionName - 既存の次元の名前です。

PropertyName - 更新するプロパティの名前です。必要なプロパティをいくつでも記載できますが、プロパティには有効な値を指定する必要があります。

Delete コマンド

サブトピック

- [Application](#)
- [Association](#)
- [Dimension](#)
- [Member](#)

Application

指定されたアプリケーションを削除します。

```
Delete Application
Properties(ApplicationName, WaitForCompletion)
Values('Comma');
```

ApplicationName - 既存のアプリケーションの名前です。

WaitForCompletion - TRUE に設定すると、バッチ・クライアントはジョブが終了するまで待機します。FALSE に設定すると、バッチ・クライアントはジョブを送信して続行します。デフォルト値は FALSE です。使用可能な値:

- TRUE
- FALSE

Association

既存の次元の関連付けを削除します。

```
Delete Association
Properties(ApplicationName, DimensionName, PropertyName)
Values('Comma', 'Scenario', 'SecurityClass');
```

ApplicationName - 既存のアプリケーションの名前です。

DimensionName - 既存の次元の名前です。

PropertyName - 更新するプロパティの名前です。必要なプロパティをいくつでも記載できますが、プロパティには有効な値を指定する必要があります。

Dimension

指定された次元を削除します。

```
Delete Dimension
Properties(ApplicationName, DimensionName)
Values('Comma', 'C_Scenario');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリの次元を削除する場合は、`#Shared` を使用します。

DimensionName - 既存の次元の名前です。

Member

指定されたメンバーを削除します。メンバーのすべての子孫を削除することもできます。

```
Delete Member
Properties(ApplicationName, DimensionName, ParentName, MemberName,
DeleteAllDescendants)
Values('Comma', 'C_Account', '#root', 'TestMember1', 'false');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリの次元メンバーを削除する場合は、`#Shared` を使用します。

DimensionName - 既存の次元の名前です。

ParentName - メンバーを削除する親の名前です。ツリーのトップ・レベルのメンバーを削除する場合は、`#Root` を使用します。

MemberName - 削除するメンバーの名前です。

DeleteAllDescendants - 削除されるメンバーの下の子孫も削除するかどうかを指定します。使用可能な値:

- TRUE
- FALSE

Detach Dimension コマンド

アプリケーション内の共有次元をローカル次元に変換します。

```
Detach Dimension
Properties(ApplicationName, DimensionName, RetainFilteredStructure,
RetainPropertyOverrides, waitForCompletion)
Values('Comma', 'Period', 'true', 'true', 'true');
```

ApplicationName - 既存のアプリケーションの名前です。

DimensionName - 既存の次元の名前です。

RetainFilteredStructure - TRUE に設定すると、次元の現在のビューが維持され、除外されたメンバーまたは他のフィルタ・メンバーは次元のローカル・コピーに表

示されません。FALSE に設定すると、次元のすべてのメンバーが次元のローカル・コピーに表示されます。使用可能な値:

- TRUE
- FALSE

RetainPropertyOverrides - TRUE に設定すると、すべてのプロパティの上書きが保持されます。それ以外の場合は、次元の共有バージョンからの値が使用されます。使用可能な値:

- TRUE
- FALSE

WaitForCompletion - TRUE に設定すると、バッチ・クライアントはジョブが終了するまで待機します。FALSE に設定すると、バッチ・クライアントはジョブを送信して続行します。使用可能な値:

- TRUE
- FALSE

Exclude コマンド

Member

共有次元からメンバーを除外します。

```
Exclude Member
  Properties(ApplicationName, DimensionName, ParentName, MemberName)
  Values('Comma', 'Period', '#root', 'P1');
```

ApplicationName - 既存のアプリケーションの名前です。次元を除外するための目標として#Sharedを使用できません。

DimensionName - 共有ライブラリ内の既存の次元の名前で、除外する次元です。

ParentName - 除外するメンバーの親の名前です。

MemberName - 除外するメンバーの名前です。

Execute コマンド

サブトピック

- [インポート\(アプリケーションまたは共有ライブラリへ\)](#)
- [次元の同期\(共有ライブラリとの\)](#)
- [配置](#)
- [再配置](#)
- [データの同期](#)
- [検証](#)

EXECUTE コマンドを使用して、ジョブを実行できます。

インポート(アプリケーションまたは共有ライブラリへ)

既存のインポート・プロファイルを実行します。

フラット・ファイルの場合、構文は次のとおりです:

```
Execute Import
Parameters(importtype, profilename, filename, waitforcompletion)Values('flatfile',
'Comma', '.\AppFiles\CommaApp.ads', 'true');
```

インタフェース・テーブルの場合、構文は次のとおりです:

```
Execute Import
Parameters(importtype, profilename, waitforcompletion)
Values('InterfaceTables', 'intapp', 'true');
```

Data Relationship Management の場合、構文は次のとおりです:

```
Execute Import
Parameters(importtype, profilename, waitforcompletion)
Values('DRM', 'DRM Profile', 'true');
```

ImportType - 実行するインポートの種類です。使用可能な値:

- FlatFile
- InterfaceTables
- DRM (Data Relationship Management)

ProfileName - 既存のインポート・プロファイルの名前です。

FileName - フラット・ファイルのタイプをインポートする場合のインポートするフラット・ファイルの名前です。

注： インタフェース・テーブルまたは Data Relationship Management のファイル
をインポートする場合は、FileName パラメータを使用しません。これは、
フラット・ファイルをインポートする場合にのみ使用されます。

WaitForCompletion - TRUE に設定すると、バッチ・クライアントはジョブが終了
するまで待機します。FALSE に設定すると、バッチ・クライアントはジョブを送
信して続行します。使用可能な値:

- TRUE
- FALSE

次元の同期(共有ライブラリとの)

共有ライブラリと次元の同期をやり取りします。

```
Execute Dimensionsynchronization  
Parameters(SourceApplicationName, DestApplicationName, DestDimensionName,  
ReplaceMode)  
Values('appName', 'DestAppName', 'DimName', 'true');
```

ApplicationName - 既存のアプリケーションの名前です。次元の同期のアプリケー
ション名として#Shared を使用できません。

DimensionName - アプリケーションの既存の次元の名前です。

SharedDimensionName - 共有ライブラリ内の既存の次元の名前です。

SyncToApp - TRUE 値を指定すると、共有次元はアプリケーションと同期され、
FALSE 値を指定すると、アプリケーションが共有ライブラリに同期されます。使
用可能な値:

- TRUE
- FALSE

ReplaceMode - TRUE 値を指定すると、次元の同期は置換モードを使用し、FALSE
値を指定すると、マージ・モードを使用します。使用可能な値:

- TRUE
- FALSE

配置

アプリケーションを指定された製品に配置します。

```
Execute Deploy  
Parameters(ApplicationName, InstanceName, ApplicationServer, HubProject, ClearAll,  
CheckIntegrity, waitforcompletion, purgeTransactions, deployOption, Notes)  
Values('AppName', 'HFM931', 'localhost:1423', 'Default Application Group', 'false',  
'false', 'true', 'true', 'AppView', 'deploy comments sample');
```

ApplicationName - 既存のアプリケーションの名前です。配置のアプリケーション名として#Sharedを使用できません。

InstanceName - 配置先のインスタンスの名前です。

ApplicationServer - 配置先のアプリケーション・サーバーの名前です。

HubProject - 配置済アプリケーションを追加する Oracle Hyperion Shared Services プロジェクトです。

ClearAll - アプリケーション・データベース内のすべての次元メンバー、および対応するデータ、仕訳、内部取引を削除します。(連結アプリケーションでのみ使用できます。)使用可能な値:

- TRUE
- FALSE

CheckIntegrity - メタデータとデータをチェックして、整合性を確認します。(連結アプリケーションでのみ使用できます。)使用可能な値:

- TRUE
- FALSE

PurgeTransactions - トランザクション履歴を削除します。TRUE 値では履歴の削除が行われますが、FALSE 値では行われません。使用可能な値:

- TRUE
- FALSE

deployOption - 実行する配置のタイプを指定します。デフォルト値は AppView で、アプリケーションのみ配置します。

WaitForCompletion - TRUE に設定すると、バッチ・クライアントはジョブが終了するまで待機します。FALSE に設定すると、バッチ・クライアントはジョブを送信して続行します。使用可能な値:

- TRUE
- FALSE

Planning アプリケーション向け:

datasourceName - 指定した名前でデータ・ソースを作成します。この値は文字列です。

CreateOutline - 初めてアプリケーションを配置する場合は、Essbase アウトラインを作成します。使用可能な値:

- TRUE
- FALSE

RefreshOutline - アプリケーションの構造を変更した後にアプリケーション・データベースをリフレッシュします。使用可能な値:

- TRUE
- FALSE

CreateSecurityFilters - 暗号化されたデータ・ファイルにアクセス権を保管します (Essbase.sec)。使用可能な値:

- TRUE
- FALSE

SharedMembersSecurityFilters - アクセス権を共有メンバーに適用します。使用可能な値:

- TRUE
- FALSE

ValidateSecurityFilterLimit - 1 行当たり 64KB の Oracle Essbase セキュリティ・フィルタの限界を超えるセキュリティ・フィルタを識別します。これによって、Oracle Essbase のセキュリティ・フィルタを構築する前に、フィルタ・サイズを検証してサイズの限界を超えないようにできます。使用可能な値:

- TRUE
- FALSE

注意 - オプションで、配置に関して追加するコメントや注意事項を保管します。

再配置

アプリケーションを製品サーバーに再配置します。

```
Execute Redeploy
```

```
Parameters(ApplicationName, InstanceName, HubProject, ClearAll, CheckIntegrity,  
waitforcompletion, purgeTransactions, deployOption, escapeValidateRules, Notes)  
Values('AppName', 'HubProj', 'false', 'false', 'true', 'true', 'AppView', 'true',  
'Comments on redeployment');
```

有効値については、[30 ページの「配置」](#)の例を参照してください。

アプリケーションを再配置する場合、`deployOption` で指定できる値は、Oracle Hyperion Planning および Oracle Hyperion Financial Management アプリケーションを配置する場合の値とは異なります。使用可能な値:

- AppView - アプリケーションのみ配置します。(この設定はデフォルトです。)
- Rules - 計算ルールを配置します。
- All - 計算ルールとアプリケーションを配置します。

他のすべてのアプリケーション・タイプでは、AppView のみ指定できます。

データの同期

既存のデータ同期プロファイルを実行します。

```
Execute DataSynchronization
Parameters (DataSynchronizationName, DataTransformationOperator,
DataTransformationValue, FileName, ValidateOnly, WaitForCompletion)
Values ('CommaSync3', '*', '1.2345', '', 'false', 'true');
```

DataSynchronizationName - 実行するデータ同期プロファイルの名前です。

DataTransformationOperator - 指定できる値は次のとおりです:

- なし
- '*' (乗算)
- '/' (除算)
- '+' (加算)
- '-' (減算)

DataTransformationValue - データ値を変更するために DataTransformationOperator と連携して使用する値です。

FileName - 同期で、同期のソースに外部のソース・ファイルを使用する場合、外部ソース・ファイルの場所です。場所は、Web サーバーがアクセスできる場所を URL で示す必要があります。ファイル・パスは、ファイルがリモート・コンピュータ上にある場合は「file:/// <machine name> / <folder name> / <file name>」、ファイルがデータの同期サービスが実行されているコンピュータ上にある場合は「file:/// C: / CvgCompLd_JobTtl.txt」である必要があります。

ValidateOnly - データの同期を実行せずに検証します。

WaitForCompletion - TRUE に設定すると、バッチ・クライアントはジョブが終了するまで待機します。FALSE に設定すると、バッチ・クライアントはジョブを送信して続行します。使用可能な値:

- TRUE
- FALSE

これらのパラメータの他に、データ・ロード・オプション・パラメータもあります。データ・ロード・オプション・パラメータは次のとおりです:

- 動的
- レジストリから取得
- パラメータ名がアプリケーションのタイプに基づいて異なる
- 他のパラメータと異なり、すべてのパラメータが大文字と小文字を区別する
- すべての値が大文字と小文字を区別する

Planning の場合、パラメータは次の 1 つです: `dataSyncLoadOptionHpMode`。使用可能な値:

- ADD
- SUBTRACT
- OVERWRITE

Essbase ASO の場合、パラメータは次の 1 つです: `dataSyncLoadOptionAsoMode`。使用可能な値:

- ADD
- SUBTRACT
- OVERWRITE

Essbase BSO の場合、パラメータは次の 1 つです: `dataSyncLoadOptionBsoMode`。使用可能な値:

- ADD
- SUBTRACT
- OVERWRITE

Planning、Essbase ASO および Essbase BSO アプリケーションの場合、デフォルト値は `OVERWRITE` です。

連結の場合、次の 2 つのパラメータがあります:

- `dataSyncLoadOptionHfmMode` - 使用可能な値は次のとおりです:
 - merge
 - replace
 - accumulate

デフォルト値は `merge` です。

- `dataSyncLoadOptionHfmAccumulateInFile` - 使用可能な値は次のとおりです:
 - TRUE
 - FALSE

デフォルトは `false` です。

注: Profitability アプリケーションおよび Profitability-詳細アプリケーションでは、データ・ロード・オプションはサポートされません。

検証

アプリケーションの検証を実行します。StopOnError オプションが TRUE に設定されているかぎり、スクリプトは検証が失敗した場合に終了します。

```
Execute Validate
Parameters(ApplicationName, ValidateType)
  Values('Comma1', 'All');
```

ApplicationName - 既存のアプリケーションの名前です。

ValidateType - 実行する検証のタイプです。たとえば、アプリケーションのみの検証、ビジネス・ルールの検証またはすべて(アプリケーションおよびルール)の検証があります。使用可能な値:

- AppView
- Rules
- All

Include コマンド

Dimension

共有ライブラリから指定したアプリケーションに既存の次元を追加します。次元は共有次元として追加することも、ローカル次元としてアプリケーションにコピーすることもできます。

```
Include Dimension
Properties(DimensionName, IncludeAsShared)
Values('C_Alias', 'true');
```

ApplicationName - 既存のアプリケーションの名前です。次元を含めるための目標として#Sharedを使用できません。

DimensionName - 共有ライブラリ内の既存の次元の名前で、インクルードする次元です。

IncludeAsShared - 共有ライブラリのソース次元とのリンクを維持する共有次元として次元をインクルードする場合は TRUE 値を使用します。共有ライブラリの次元とは別に次元のコピーをインクルードする場合は FALSE 値を使用します。使用可能な値:

- TRUE
- FALSE

Insert コマンド

Insert Member

メンバーのコピーを共有メンバーとして挿入します。Insert Member コマンドを使用できるのは、ローカル次元および共有ライブラリ次元のみです。アプリケーション内の共有次元ではメンバーを挿入できません。

```
Insert Member Properties(DimensionName, ParentName, InsertMemberName,
MemberToInsertName)
Values('Account', 'Par1', 'Mem1', 'Mem2');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリの次元メンバーを操作する場合は、#Shared を使用します。

DimensionName - 既存の次元の名前です。

ParentName - メンバーの挿入先の親の名前です。

InsertMemberName - メンバーの挿入先のメンバーの名前です。

MemberToInsertName - 挿入する共有メンバーの名前です。

移動コマンド

メンバーの移動

次元構造内のある場所から別の場所にメンバーを移動します。Move Member コマンドを使用できるのは、ローカル次元および共有ライブラリ次元のみです。アプリケーション内の共有次元のメンバーは移動できません。

```
Move Member
Properties(ApplicationName, DimensionName, FromParentName,
MemberName, ToParentName, InsertAfterMemberName)
Values('SampleApp', 'Period', '#root', 'r1', 'P2', '#none');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリの次元メンバーを操作する場合は、#Shared を使用します。

DimensionName - 既存の次元の名前です。

FromParentName - 移動するメンバーの親の名前です。

MemberName - 移動するメンバーの名前です。

ToParentName - メンバーの移動先の親の名前です。

InsertAfterMember - メンバーの挿入先となる ToParentName に属する子を示します。メンバーを子として挿入する必要があることを示す特別な値'#none'を設定で

きます。これにより、挿入されるメンバーとその後のすべてのメンバーのソート順が影響を受けます。

Rename コマンド

メンバー名の変更

メンバーおよびそのメンバーのすべての共有コピーを名前変更します。Rename Member コマンドを使用できるのは、ローカル次元および共有ライブラリ次元のみです。アプリケーション内の共有次元のメンバーは名前変更できません。

```
Rename Member
  Properties(ApplicationName, DimensionName, ParentName, MemberName, NewMemberName)
  Values('Comma', 'Account', '#root', 'M2', 'M2REN');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリの次元メンバーを操作する場合は、#Shared を使用します。

DimensionName - 既存の次元の名前です。

ParentName - 名前変更するメンバーの親の名前です。

MemberName - 名前変更する既存のメンバーの名前です。

NewMemberName - メンバーの新しい名前です。

Remove コマンド

サブトピック

- [Remove Member \(アプリケーションまたは共有ライブラリ\)](#)
- [Remove Dimension](#)

Remove Member (アプリケーションまたは共有ライブラリ)

指定した次元からメンバーを除去しますが削除はしません。Remove Member コマンドを使用できるのは、アプリケーション内の共有次元を除去する場合のみです。

```
Remove Member
  Properties(DimensionName, ParentName, MemberName)
  Values('Account', 'Mem1', 'Mem2');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリの次元メンバーを操作する場合は、#Shared を使用します。

DimensionName - 既存の次元の名前です。

ParentName - 除去するメンバーの親の名前です。

MemberName - 除去するメンバーの名前です。

Remove Dimension

アプリケーションから共有次元を除去します。

```
Remove Dimension
  Properties(ApplicationName, DimensionName, Force)
  Values('Comma', 'Period', 'true');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリの次元メンバーを操作する場合は、#Shared を使用します。

DimensionName - 既存の次元の名前です。

Force - TRUE に設定した場合、除去対象の次元がアプリケーション内の他の次元に関連付けられていても除去されます。FALSE に設定した場合、除去対象の次元がアプリケーション内の他の次元に関連付けられていると次元を除去できません。使用可能な値:

- TRUE
- FALSE

Share Dimension コマンド

ローカル次元を新しい共有次元に変換するか、既存の次元とマージします。

```
Share Dimension
  Properties(ApplicationName, DimensionName, ShareAsNew, SharedDimensionName,
MergeAsShared,
  WaitForCompletion)
  Values('Comma', 'Entity', 'false', 'ShareEntity', 'true', 'true');
```

ApplicationName - 既存のアプリケーションの名前です。

DimensionName - 既存の次元の名前です。

ShareAsNew - TRUE に設定すると、SharedDimensionName プロパティおよび MergeAsShared プロパティが無視されます。これは、次元を新規として共有する場合に、これらのプロパティが適用されないためです。使用可能な値:

- TRUE
- FALSE

SharedDimensionName - FALSE に設定する場合は、共有ライブラリで共有する次元の名前を指定する必要があります。使用可能な値:

- TRUE
- FALSE

MergeAsShared - TRUE に設定すると、共有対象の次元がターゲット次元とマージされます。FALSE に設定すると、共有対象の次元によってターゲット次元が置換されます。使用可能な値:

- TRUE
- FALSE

WaitForCompletion - TRUE に設定すると、バッチ・クライアントはジョブが終了するまで待機します。FALSE に設定すると、バッチ・クライアントはジョブを送信して続行します。使用可能な値:

- TRUE
- FALSE

Update コマンド

サブトピック

- [Application](#)
- [Dimension](#)
- [Member](#)
- [次元の関連付け](#)

UPDATE スクリプト・コマンドを使用して、アプリケーション、次元またはメンバーのプロパティ値を変更する場合は、Performance Management Architect のプロパティ・グリッドに表示されるプロパティ・ラベルではなくプロパティ名を使用する必要があります。プロパティ・ラベルとプロパティ名については、『Oracle Hyperion Enterprise Performance Management Architect 管理者ガイド』の付録に説明があります。次に、スクリプトの例を示します:

```
Update Member
Properties(DimensionName, ParentName, MemberName, AggregationWeight,
NumDecimalPlaces)
Values('ScenarioDim', '#root', 'Member1', '3', '2');
```

この例では、DimensionName、ParentName および MemberName は、すべて標準のスクリプト・アイテムですが、AggregationWeight と NumDecimalPlaces は、Oracle Hyperion EPM Architect のメンバー・レベルのプロパティです。

Application

指定されたアプリケーションの 1 つ以上のプロパティを更新します。

```
Update Application
Properties(ApplicationName, ValidationAccount)
Values('Comma', 'Validation');
```

ApplicationName - 既存のアプリケーションの名前です。 #Shared のプロパティ値は更新できません。

PropertyName - 更新するプロパティの名前です。必要なプロパティをいくつでも記載できますが、プロパティには有効な値を指定する必要があります。

Dimension

指定された次元の 1 つ以上のプロパティを更新します。

```
Update Dimension
Properties(ApplicationName, DimensionName, PropertyName)
Values('Comma', 'C_Entity', 'Validation');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリの次元を更新する場合は、 #Shared を使用します。

DimensionName - 既存の次元の名前です。

PropertyName - 更新するプロパティの名前です。必要なプロパティをいくつでも記載できますが、プロパティには有効な値を指定する必要があります。

Member

指定されたメンバーの 1 つ以上のプロパティを更新します。

```
Update Member
Properties(ApplicationName, DimensionName, ParentName, MemberName, ValidationAccount)
Values('Comma', 'C_Entity', 'E1', 'E1-1', 'Validation');
```

ApplicationName - 既存のアプリケーションの名前です。共有ライブラリの次元メンバーを更新する場合は、 #Shared を使用します。

DimensionName - 既存の次元の名前です。

ParentName - 更新するメンバーの親の名前です。

MemberName - 更新するメンバーの名前です。

PropertyName - 更新するプロパティの名前です。必要なプロパティをいくつでも記載できますが、プロパティには有効な値を指定する必要があります。

次元の関連付け

アプリケーションのタイプおよびアプリケーションに含まれる次元に基づいたすべての標準次元の関連付けをアクティブ化します。


```
Update Dimensionassociation  
Properties(activateallforapplication) Values('true');
```

ApplicationName - 既存のアプリケーションの名前です。#Sharedのプロパティ値は更新できません。

ActivateAllForApplication - 指定したアプリケーションに対してすべての関連付けをアクティブ化するかどうかを指定します。使用可能な値:

- TRUE
- FALSE

索引

記号

-C コマンド, 8
 -G コマンド, 9
 -H コマンド, 8
 -L コマンド, 9
 -O コマンド, 10
 -P コマンド, 9
 -R コマンド, 9
 -S コマンド, 9
 -U コマンド, 9
 -V コマンド, 10

A - Z

application
 create, 23
 delete, 26
 delete association, 26
 update, 39
 copy application コマンド, 11, 22
 copy dimension コマンド, 11, 22
 create
 application, 23
 dimension, 24
 member, 24
 create application コマンド, 11
 create association コマンド, 25
 create dimension association コマンド, 11
 create dimension コマンド, 11
 create member コマンド, 11
 debug コマンド, 11
 delete
 application, 26
 application association, 26
 dimension, 26
 member, 27
 delete application コマンド, 11
 delete association コマンド, 25
 delete dimension association コマンド, 11

delete dimension コマンド, 11
 delete member コマンド, 11
 detach dimension
 コマンド, 27
 detach dimension コマンド, 11
 dimension
 create, 24
 delete, 26
 include, 35
 remove, 38
 share, 38
 update, 40
 exclude member
 コマンド, 28
 exclude member コマンド, 11
 execute
 インポート, 29
 検証, 34
 再配置, 32
 次元の同期, 30
 データの同期, 33
 配置, 30
 メンバーの移動, 36
 execute data synchronization コマンド, 11
 execute deploy コマンド, 11
 execute dimension synchronization コマンド, 11
 execute import コマンド, 11
 execute redeploy コマンド, 11
 execute validate コマンド, 11
 exit コマンド, 11, 22
 include
 dimension, 35
 include dimension コマンド, 11
 include member コマンド, 11
 insert
 member, 36
 insert コマンド, 11
 login コマンド, 12, 20

logout コマンド, 12, 21

member

create, 24

delete, 27

insert, 36

remove, 37

update, 40

move コマンド, 12

option コマンド, 12

quit コマンド, 12, 22

remove

dimension, 38

member, 37

remove dimension コマンド, 12

remove member コマンド, 12

rename コマンド, 12

set コマンド, 12

share

dimension, 38

share dimension コマンド, 12

update

application, 39

dimension, 40

member, 40

コマンド, 39

次元の関連付け, 40

update application コマンド, 12

update association コマンド, 12

update dimension コマンド, 12

update member コマンド, 12

variable コマンド, 12

create association, 25

create dimension, 11

create dimension association, 11

create member, 11

debug, 11

delete application, 11

delete association, 25

delete dimension, 11

delete dimension association, 11

delete member, 11

detach dimension, 11

exclude member, 11, 28

execute data synchronization, 11

execute deploy, 11

execute dimension synchronization, 11

execute import, 11

execute redeploy, 11

execute validate, 11

exit, 11, 22

include dimension, 11

include member, 11

insert, 11

login, 12, 20

logout, 12, 21

move, 12

option, 12

quit, 12, 22

remove dimension, 12

remove member, 12

rename, 12

set, 12

share dimension, 12

update, 39

update application, 12

update association, 12

update dimension, 12

update member, 12

variable, 12

-C, 8

-G, 9

-H, 8

-L, 9

-O, 10

-P, 9

-R, 9

-S, 9

-U, 9

あ行

一般エラーのコマンド, 11

インポート

execute, 29

か行

解析エラーのコマンド, 11

検証

execute, 34

検証エラーのコマンド, 11

コマンド, 18

application, 19

copy application, 11, 22

create application, 11

-V, 10

一般エラー, 11

一般的, 19

解析エラー, 11

検証エラー, 11

コマンド・ライン・エラー, 11

次元, 19

次元のコピー, 22

ジョブの実行, 29

正常終了, 11

メンバー, 19

メンバー名の変更, 37

コマンド・ファイル

構成, 15

コマンド, 18

コメント, 17

スクリプト, 19

変数, 16

コマンド・ライン・エラー, 11

コマンド・ラインのオプション, 8

コマンド・ライン・モード, 7

コメント, 17

さ行

再配置

execute, 32

終了条件, 10

次元の関連付け

update, 40

次元の同期

execute, 30

スクリプト, 19

スクリプト・モード, 7

正常終了のコマンド, 11

た行

データの同期

execute, 33

は行

配置

execute, 30

バッチ・クライアント

起動, 8

使用, 7

情報, 7

変数, 16

ま行

メンバーの移動

execute, 36

メンバー名の変更

コマンド, 37

戻りコード, 10

ら行

ロギング, 12

